Serdica J. Computing **14** (2020), No 1–2, 27–41

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

SOME NP-COMPLETE PROBLEMS FOR ATTRIBUTE REDUCTION IN CONSISTENT DECISION TABLES

Phan Dang Khoa, János Demetrovics, Vu Duc Thi, Pham Viet Anh

Abstract. Over recent years, the research of attribute reduction for general decision systems and, in particular, for consistent decision tables has attracted great attention from the computer science community due to the emerge of big data. It has been known that, for a consistent decision table, we can derive a polynomial time complexity algorithm for finding a reduct. In addition, finding redundant properties can also be done in polynomial time. However, finding all reduct sets in a consistent decision table is a problem with exponential time complexity. In this paper, we study complexity of the problem for finding a certain class of reduct sets. In particular, we make use of a new concept of relative reduct in the consistent decision table. We present two NP-complete problems related to the proposed concept. These problems are related to the cardinality constraint and the relative reduct set. On the basis of this result, we show that finding a reduct with the smallest cardinality cannot be done by an algorithm with polynomial time complexity.

ACM Computing Classification System (1998): I.5.2, I.2.6.

Key words: attribute reduction, NP-complete, complexity, consistent decision table, rough set theory..

1. Introduction. Reduction of attribute set on decision tables is one of the most important problems for data pre-processing in data mining and machine learning [18, 2]. The primary objective of attribute reduction in an information system is to remove redundant and unnecessary attributes in order to find a reduct attribute set (known as a *reduct*) for data storage and query efficiency. Furthermore, this procedure also greatly helps to increase the performance of knowledge extracting models, especially for big data.

Rough set theory (RST) was introduced by Zdzisław Pawlak [15] for data analysis. The mathematical theory is considered as an effective method to find reduct of decision systems. In the literature, there are a number of complexity studies and algorithms based on RST and its variants to find attribute reductions of decision systems over the last decades [10, 14, 17, 13, 3]. However, due to the hardness of the problem, most of proposed algorithms are heuristic algorithms, whose aim is to find *one* reduct to obtain a certain metric such as the best out-of-sample accuracy for prediction models [18, 19, 4, 22]. In this paper, we are interested in analyzing exact algorithms for the attribute reduction problem for a class of reduct sets in a consistent decision table under a new concept called *relative reduct*, for which we will introduce later.

On *consistent* decision systems, in recent years there have been a number of publications related to reduct of consistent decision systems according to the relational database theory approach [7, 20]. In papers [12, 20], the authors constructed an algorithm to find *all* reductive attributes of consistent decision systems in polynomial time. However, many problems related attribute reduct have more complex structure. In [11], the authors study equivalence properties of reduct of consistent decision systems related to a Sperner-system. In [9], the authors investigate the time complexity of algorithms for the inclusion problem of families of relative reducts and minimal sets in a consistent decision table. In paper [8], the authors proved that the problem of finding all reducts has the exponential time complexity in the cardinality of conditional attribute set. Therefore, state-of-the-art algorithms for finding all reducts using rough set theory might only be effective on small and medium-sized datasets, they are not scalable with big ones. We notice that inconsistent and incomplete decision systems have also been considered in the literature [21].

In this paper, by using rough set theory and relational database theory, we study the complexity for a class of problems related to attribute reduction of consistent decision systems. In particular, we argue that checking reducible property or redundancy of an attribute can be conducted in polynomial time. With the help of a new notation of relative reduct, i.e, an attribute set contains a certain reduct, we introduce two problems and prove that under a cardinality constraint they are NP-complete.

The rest of this paper is structured as follows. Section 2 reviews the preliminaries of rough set and relational database theory. In Section 3 we present some new complexity results related to attribute reduction problems. We conclude the paper in Section 4.

2. Preliminaries.

We note that the study of reductions for a consistent decision table is closely related to the relational database theory. In this section, for completeness, we review some basic concepts in the relational database theory and rough set theory, which are used in our analysis. For more details of these concepts, the reader is referred to [6, 7, 12, 20].

2.1. Relational Database Theory.

This subsection briefly reviews some key notations of the relational data model including a relation, functional dependency, f-family, and a Sperner system.

Definition 1. *Let $R = \{a_1, ..., a_n\}$ be a finite and nonempty attribute set, and $D(a_i)$ be the set of all values of attribute a_i. We say that the set of tuples $r = \{h_1, \ldots, h_m\}$ is a* ***relation*** *over R if every function $h_j : R \to \bigcup_{a_i \in R} D(a_i), 1 \leq j \leq m$, satisfies the condition $h_j(a_i) \in D(a_i)$ for all $i = 1, \ldots, n$.*

Let $r = \{h_1, \ldots, h_m\}$ be a relation over set $R = \{a_1, \ldots, a_n\}$. Functional dependency over R is denoted by $A \to B$ where A, B are attribute sets and $A, B \subseteq R$. Functional dependency satisfies the relation r over R if:

$$\Big((\forall a \in A)\big(h_i(a) = h_j(a)\big) \Rightarrow (\forall b \in B)\big(h_i(b) = h_j(b)\big)\Big)$$

for all $h_i, h_j \in r$. The set $F_r = \big\{(A, B) : A, B \subseteq R, A \to B\big\}$ is called a full family of functional dependencies in r. Let $P(R)$ be the power set of R. A family $F \subseteq P(R) \times P(R)$ is called an f-family over R if and only if for all subsets of attributes $A, B, C, D \subseteq R$ the following properties hold:

(1) $(A, A) \in F$;

(2) $(A, B) \in F, (B, C) \in F \Rightarrow (A, C) \in F$;

(3) $(A, B) \in F, A \subseteq C, D \subseteq B \Rightarrow (C, D) \in F$;

(4) $(A, B) \in F, (C, D) \in F \Rightarrow (A \cup C, B \cup D) \in F$.

We can see that F_r is an f-family over R. It has been known that if F is an f-family over R, then there is a relation r such that $F_r = F$. We denote F^+ as the functional dependency set; as a result, F^+ can be obtained from F by utilizing the rules (1) - (4).

Definition 2. *A relation schema is defined as $s = (R, F)$, where R is an attribute set and F is a functional dependency set defined on R. For any $A \subseteq R$, the closure of A on s is $A^+ = \{a : A \to \{a\} \in F^+\}$. Similarly, $A_r^+ = \{a : A \to \{a\} \in F_r\}$ is called the closure of A on relation r*

It is clear that $A \to B \in F^+$ if and only if $B \subseteq A^+$.

Definition 3. *A family $K \subseteq P(R)$ is a **Sperner system** on R if for any $A, B \in K$ implies $A \not\subset B$. Let K be a Sperner system over R, we define the set K^{-1} as follows:*

$$\begin{aligned} K^{-1} = \{ & A \in R : (B \in K) \Rightarrow (B \not\subset A) \\ & \textit{and } (A \subset C) \Rightarrow (\exists B \in K)(B \subseteq C)\} \end{aligned}$$

We say that K^{-1} is an anti-keys set.

We remark that K^{-1} is also a Sperner system. If K is a Sperner system over R as the set of all minimal keys of relation r (or relation schema s) then K^{-1} is the set of subsets of R. It does not contain the element of K and is maximal for this property. Let r be a relation on R and consider $a \in R$. We define K_a^r as follows:

$$\begin{aligned} K_a^r = \Big\{ & A \subseteq R : A \to \{a\} \in F_r, \\ & \nexists B : \big(B \to \{a\} \in F_r\big)(B \subset A)\Big\} \end{aligned}$$

Then, K_a^r is called family of minimal sets of attribute a over relation r.

2.2. Rough Set Theory. The rough set theory is particularly concerned about information systems [16]. It is a powerful tool to study attribute reduction for data analysis and knowledge acquisition.

Definition 4. *An **information system** S is an order quadruple $S = (U; A; V; f)$, where U is a finite and nonempty set of objects, called the universe; A is a finite and nonempty set of attributes; $V = \bigcup_{a \in A} V_a$ where V_a is the set of values of attribute $a \in A$; $f : U \times A \to V_a$ is the information function, such that for every $a \in A, u \in U : f(u, a) \in V_a$.*

For every $u \in U$ and $a \in A$, we denote the value of attribute a of object u as $a(u) = f(u, a)$. If $B = b_1, ..., b_k \subseteq A$ is subset of attributes, then the set of $b_i(u)$ is denoted as $B(u)$. Therefore, if u and y are two objects in U, then $B(u) = B(v)$ if and only if $b_i(u) = b_i(v)$ for all $i = 1, \ldots, k$.

Definition 5. *A **decision table** is an information system $S = (U; A; V; f)$, where C is the condition attribute set, D is the decision attribute set, $A = C \cup D$ and $C \cap D = \emptyset$. A decision table S is **consistent** if the functional dependency $C \to D$ is true; that is, for every $u, v \in U$ if $C(u) = C(v)$ then $D(u) = D(v)$. Otherwise, decision table S is **inconsistent**.*

Without loss of generality, we assume that D consists of only one decision attribute d. From now on, we can analyze a decision table of the from $DS = (U, C \cup \{d\}, V, f)$. A formal definition for a reduct is

Definition 6. *For a consistent decision table $DS = (U, C \cup \{d\}, V, f)$ and an attribute set $B \subseteq C$, B is called a **reduct** of C if:*

(1) for every $u, v \in U$, if $B(u) = B(v)$ then $d(u) = d(v)$;

(2) for all $E \subset B$, there exist $u, v \in U$ such that $E(u) = E(v)$ and $d(u) \neq d(v)$.

For convenience, we let $PRED(C)$ denote the set of all reducts of C.

2.3. Vertex Cover Set and a Polynomial Time Algorithm for Finding the Closure. In this subsection we present the definition of vertex cover set [1], an algorithm for finding the closure of attribute set [6, 8], and a theorem about minimal sets of attribute over relation [8], which we will further need for our complexity analysis in Section 3.

Definition 7. *(Vertex cover set) Given an undirected graph $G = (V, E)$, where V is the set of vertices and E is the set of edges, the set $C \subseteq V$ is a vertex cover of G if $C \cap \{a_i, a_j\} \neq \emptyset$ for every edge $(a_i, a_j) \in E$.*

In [6], we introduced an algorithm to find the closure of attribute set, which can be summarized as follows:

We can see the time complexity of this algorithm is polynomial with respect to n and m. Furthermore, it is clear that $A \to B \in F_r$ *in relation* r if and only if $B \subseteq A_r^+$. Furthermore, let us recall the following result characterizing a family of minimal sets, which was proved in [8].

Algorithm 1: Finding the closure of attribute set [6, 8]

Input: $r = \{h_1, \dots, h_m\}$ is a relation over $R = \{a_1, \dots, a_n\}$ and $A \subseteq R$

Output: A_r^+ is the closure of A on relation r

1 Step 1: From r build $E_r = \{E_{ij} : 1 \leq i < j \leq m\}$ where $E_{ij} = \big\{a \in R : h_i(a) = h_j(a)\big\}$.

2 Step 2: Build $M = \{B : \exists E_{ij}, B = E_{ij}\}$

3 Step 3: Set $A_r^+ = \begin{cases} \bigcap B, & if \;\; \exists B \in M : A \subseteq B \\ R & otherwise \end{cases}$

Theorem 1. *[8] Assume that* $DS = \big(U, C \cup \{d\}, V, f\big)$ *is a consistent decision table, where* $C = \{c_1, \dots, c_n\}$ *and* $U = \{u_1, \dots, u_m\}$. *Consider the relation* $r = \{u_1, \dots, u_m\}$ *defined on attribute set* $R = C \cup \{d\}$. *We set* $E_r = \{E_{ij} : 1 \leq i < j \leq m\}$ *where* $E_{ij} = \big\{a \in R : a(u_i) = a(u_j)\big\}$ *and* $M_d = \big\{A \in E_r : d \notin A, \not\exists B \in E_r : d \notin B, A \subset B\big\}$. *Then we have* $M_d = \big(K_d^r\big)^{-1}$, *where* K_d^r *is a family of minimal sets of attribute* d *over relation* r.

3. Complexity Analysis for Attribute Set Reduction Problems. In many applications, decision tables often contain inconsistent objects which have the same values on the conditional attributes, but different values on the decision attribute. These decision tables are called inconsistent decision tables. However, depending on the class of problems, we can convert the inconsistent decision table to a consistent decision table through a data pre-processing step by removing inconsistent objects.

It can be seen that, in any decision table DS, if we do not allow two or more rows to have the same values, then checking whether DS is a consistent decision table can be done by a polynomial-time algorithm with respect to the size of this table.

In this section, we analyze the computational complexity for two classes of problems: 1) checking reducible/redundant property of an attribute, and 2) the existence of a relative reduct of a decision table DS.

3.1. Polynomial Time Solvable for Attribute Checking Problems. In this subsection, we investigate the complexity of two fundamental problems: determining whether an attribute a is either a reduct attribute or redundant attribute. We show that checking the property can be done in polynomial time.

In [5], we propose an algorithm for finding a reduct from a consistent decision table. The statement of algorithm is given in Algorithm 2.

Algorithm 2: Finding a reduct over consistent decision table [5]

Input: $DS = \big(U, C \cup \{d\}, V, f\big)$, where
$POS_c\big(\{d\}\big) = U, C = \{c_1, \ldots, c_n\}, U = \{u_1, \ldots, u_m\}$

Output: H is the reduct

1 Let consider relation $r = \{u_1, \ldots, u_m\}$ on attribute set $R = C \cup \{d\}$.

2 Step 1: Compute $E_r = \{E_{ij} : 1 \leq i < j \leq\}$ where
$E_{ij} = \big\{a \in R : a(u_i) = a(u_j)\big\}$.

3 Step 2: From E_r compute
$M_d = \{A \in E_r : d \notin A, \nexists B \in E_r : d \notin B, A \subset B\}$.

4 Step 3: Set $L(0) = C$

5 Step $i+1$: Set

$$L(i+1) = \begin{cases} L(i) - a_{i+1}, & if \ \nexists A \in M_d : \{L(i) - a_{i+1}\} \subseteq A \\ L(i) & otherwise \end{cases}$$

6 Then we set $H = L(n)$.

As proved in [5], the number of steps computing E_r is bounded from above by $|U|^2$. The number of steps computing M_d is less than $|E_r|^2$ and $|E_r| \leq \dfrac{|U|(|U|-1)}{2}$. Thus, we have the worst-case time complexity of Algorithm 2 is not greater than $\mathcal{O}(n.m^2)$. As a result, this algorithm has a polynomial complexity.

In the following example, we illustrate how to construct a reduct by using Algorithm 2.

Example 1. *Suppose we are given a consistent decision table* $DS_1 = \big(U, C \cup \{d\}, V, f\big)$ *where* $U = \{u_1, u_2, u_3, u_4, u_5\}$ *and* $C = \{a, b, c\}$ *as shown in Table 1. We wish to find a reduct for this decision table.*

We can see that $E_{12} = c, E_{13} = ac, E_{14} = ab, E_{15} = ad, E_{23} = bcd, E_{24} = d, E_{25} = b, E_{34} = ad, E_{35} = ab$, and $E_{45} = ac$.

So we have $E_r = \{c, ac, ab, ad, bcd, d, b, ad, ab, ac\}$,
$M_d = \{ac, ab\}$ and
$L(0) = abc$.
Using Algorithm 2, it follows that:
$L(1) = bc, L(2) = bc$ and $L(3) = bc$.
So we have $H = \{b, c\}$ is a reduct of decision table DS_1.

Table 1. Consistent decision table DS_1.

	a	b	c	d
u_1	1	0	1	1
u_2	0	1	1	0
u_3	1	1	1	0
u_4	1	0	0	0
u_5	1	1	0	1

An efficient algorithm for obtaining the set of all attribute reduction is presented in Algorithm 3 [12].

Algorithm 3: Finding the set of all reduct attributes of C [12]

Input: $DS = \big(U, C \cup \{d\}, V, f\big)$, where
$POS_c\big(\{d\}\big) = U, C = \{c_1, \ldots, c_n\}, U = \{u_1, \ldots, u_m\}$

Output: $REAT(C)$ is the set of all reduct attributes of C

Let consider relation $r = \{u_1, \ldots, u_m\}$ on attribute set $R = C \cup \{d\}$.

Step 1: Compute $E_r = \{E_{ij} : 1 \leq i < j \leq\}$ where
$E_{ij} = \big\{a \in R : a(u_i) = a(u_j)\big\}$.

Step 2: From E_r compute
$M_d = \{A \in E_r : d \notin A, \nexists B \in E_r : d \notin B, A \subset B\}$.

Step 3: Construct set $N = R - \bigcap\limits_{K \in M_d} K$

Step 4: Set $REAT(C) = N - \{d\}$.

We can see that the number of computational steps for E_r is not greater than $|U|^2$ and for M_d is not greater than $|E_r|^2$. Thus, this is a polynomial time algorithm for the problem with respect to number of rows and columns of decision table DS. From the complexity analysis of Algorithm 3 we have following corollaries.

Corollary 1. *Given a consistent decision table $DS = \big(U, C \cup \{d\}, V, f\big)$ and attribute $\mathfrak{a}$, let us consider the problem of checking whether $\mathfrak{a}$ is a reduct attribute or not. Then, the problem can be solved in polynomial time with respect to the number of rows and columns of decision table DS.*

A similar polynomial time solvable result for redundant attribute verification is as follows:

Table 2. Consistent decision table DS_2.

	a	b	c	d	e	f	g
u_1	1	1	0	0	1	1	0
u_2	1	1	1	1	0	0	1
u_3	0	0	0	1	1	0	1
u_4	0	1	1	0	0	1	0

Corollary 2. *Given a consistent decision table* $DS = \big(U, C \cup \{d\}, V, f\big)$ *and attribute* $\mathtt{a}$*, let us consider the problem of checking whether* $\mathtt{a}$ *is a redundant attribute or not. Then, the problem can be solved in polynomial time with respect to the number of rows and columns of decision table* DS*.*

In the following example, we illustrate how to find the set of all reduct attributes by using Algorithm 3.

Example 2. *Consider a consistent decision table* $DS_2 = \big(U, C \cup \{d\}, V, f\big)$ *where* $U = \{u_1, u_2, u_3, u_4\}$ *and* $C = \{a, b, c, e, f, g\}$ *as shown in Table 2. We want to identify all reduct attributes.*

We can see that: $E_{12} = ab, E_{13} = cf, E_{14} = begd, E_{23} = egd, E_{24} = bcf$, and $E_{34} = a$. Hence we have $E_r = \{ab, cf, begd, egd, bcf, a\}$.

It is clear that $M_d = \{begf, bcf, ab\}$.

Therefore, we have $\{begf\} \cap \{bcf\} \cap \{ab\} = \{b\}$.

It follows that $N = R - \{b\} = \{acefgd\}$; hence we get $REAT(C) = N - \{d\} = \{acefg\}$. So finally we have $\{a, c, e, f, g\}$ is the set of all reduct attributes of the decision table in our example.

We notice that Algorithm 2 allows us to find one reduct of any consistent decision table DS in polynomial time. However finding all reducts of DS is exponential time. We have the following theorem.

Theorem 2. *[8] Given a consistent decision table* $DS = \big(U, A = C \cup \{d\}, V, f\big)$*, finding all reducts* $PRED(C)$ *of* DS *is a problem which has exponential time complexity by size of* A*.*

In order to solve this problem, we need to show two things:

1) There is an exponential time algorithm for finding $PRED(C)$;

2) There is no algorithm for finding $PRED(C)$ in less than exponential time.

We characterize the relationship between a Sperner system K and $(K_d^r)^{-1}$.

Lemma 1. *Suppose that K is a Sperner system on C, then there exists a consistent decision table $DS = \big(U, C \cup \{d\}, V, f\big)$ such that $K = (K_d^r)^{-1}$.*

Proof. Assume that $K = \{A_1, \ldots, A_m\}$. We construct decision table $DS = \big(U, C \cup \{d\}, V, f\big)$ by the following approach:

$U = \{u_0, u_1, \ldots, u_m\}$, for every $c \in C$: $c(u_0) = 0$ and $d(u_0) = 0$. For every $i \in \{1, \ldots, m\}$ and $c \in C$, we set $c(u_i) = 0$ if $c \in A_i$, otherwise $c(u_i) = i$. Set $d(u_i) = i$. We have $R = C \cup \{d\}$.

Set $E_r = \{E_{ij} : 1 \leq i < j \leq m\}$, where $E_{ij} = \big\{a \in R : a(u_i) = a(u_j)\big\}$.

Let us define $M_d = \{A \in E_r : d \notin A, \nexists B \in E_r : d \notin B, A \subset B\}$. We can see that $M_d = \{A_1, A_2, \ldots, A_m\}$. By applying Theorem 1, we have $M_d = \big(K_d^r\big)^{-1}$. It implies that $K = \big(K_d^r\big)^{-1}$. The proof is complete. □

3.2. Two NP-Complete Problems. First we show that checking the existence of following set with cardinality constraint is NP-complete.

Theorem 3. *The following problem is NP-complete.*

Given a Sperner system K on $R = \{a_1, a_2, \ldots, a_n\}$ and a positive integer $k \leq n$, let us determine whether there exists a set $A \subseteq R$ such that $|A| \leq k$ ($|A|$ denotes the cardinality of set A) and for every $B \in K$: $A \not\subset B$.

Proof. Select a set A randomly such that $|A| \leq k$ and A is not a subset of each $B \in K$. We can see that the selection problem has polynomial time complexity of n and m (where $|K| = m$). So we have an NP problem.

We consider the NP-complete problem in [1], i.e., the cardinality vertex cover problem:

For a given undirected graph $G = (V, E)$, where V is the set of vertices and E is the set of edges, and an positive integer k, let us find a vertex cover set whose cardinality is not greater than k.

We need to show how to convert this problem to our problem by a transformation in polynomial time. Assume that $G = (V, E)$ is an undirected graph and $k \leq |A|$. Let us denote $R = V$ and $P = \big\{R \setminus \{a_i, a_j\} : (a_i, a_j) \in E\big\}$. It is plain to see that P is a Sperner system on R.

Suppose that $P = \{B_1, \ldots, B_m\}$. If $|A| \leq k$ and $A \not\subset B_i, \forall i = 1, \ldots, m$, then by the definition of P we have $A \cup \{a_i, a_j\} \neq \emptyset$ for any $(a_i, a_j) \in E$. Hence, A is a vertex cover set of G. Otherwise, if A is a vertex cover set of G then from the definition of P and vertex cover set, we have A is not a subset of B_i, for every $i = 1, \ldots, m$. It implies that A is not a subset of B_i (for every $i = 1, \ldots, m$) if and only if A is a vertex cover set of G. The theorem is proved. □

Now we introduce a new notion of reduction for a consistent decision table DS. The definition of relative reduct is given as follows.

Definition 8. *Given a consistent decision table* $DS = \big(U, C \cup \{d\}, V, f\big)$, *a set* B *is called* ***relative reduct*** *of* DS *if there exists a reduct set* A *such that* $A \subseteq B$.

Example 3. *Suppose we are given a consistent decision table* $DS_3 = \big(U, C \cup \{d\}, V, f\big)$ *where* $U = \{u_1, u_2, u_3, u_4, u_5\}$ *and* $C = \{a, b, c, e\}$ *as shown in Table 3. We will find relative reducts for this decision table.*

Table 3. Consistent decision table DS_3.

	a	b	c	e	d
u_1	1	0	1	1	1
u_2	0	1	1	0	0
u_3	1	1	1	1	0
u_4	1	0	0	0	0
u_5	1	1	0	1	1

We can see that: $E_{12} = c, E_{13} = ace, E_{14} = ab, E_{15} = aed, E_{23} = bcd, E_{24} = ed, E_{25} = b, E_{34} = aed, E_{35} = abe$, and $E_{45} = ac$. So we have:

$E_r = \{c, ace, ab, aed, bcd, ed, b, aed, abe, ac\}$;
$M_d = \{ace, abe\}$;
$L(0) = \{abce\}$.

Using Algorithm 2, it follows that:

$L(1) = abc$;
$L(2) = bc$;
and $L(3) = bc$;

So we have $A = \{bc\}$ is a reduct of decision table DS_3. Following Definition 8, we conclude that DS_3 has two relative reducts with cardinality 3: $B_1 = \{abc\}$ and $B_2 = \{bce\}$.

Based on Lemma 1, we have an algorithm with polynomial time complexity for finding a consistent decision table from a Sperner-system K satisfying $K_d^{-1} = K$. Therefore, when applying this result, we can derive the following complexity result.

Theorem 4. *The following problem is NP-complete.*

Given a positive integer k ($k \leq |C|$) and a consistent decision table $DS = \big(U, C \cup \{d\}, V, f\big)$, let us determine whether there exists a relative reduct *set A of DS such $|A| \leq k$.*

Proof. We select A randomly such that $|A| \leq k$. By using Algorithm 1, we determine $A \to \{d\}$. We know that $A \to \{d\}$ if and only if $\{d\} \subseteq A_r^+$. From Algorithm 1, this determination has polynomial time complexity. Hence, the above problem is NP.

We select the problem in Theorem 3 as an NP-complete problem:

Given a Sperner system K on $R = \{a_1, ..., a_n\}$ and a positive integer k ($k \leq n$). Determining whether there exists a set $A \subseteq R$ such that $|A| \leq k$ and for any B ($B \in K$): $A \not\subset B$.

We prove that this problem can be converted into our problem by a transformation with polynomial time.

Define $K = \{B_1, \ldots, B_m\}$, we construct the decision table $DS = \big(U, C \cup \{d\}, V, f\big)$ as follows:

$U = \{u_0, u_1, \ldots, u_m\}$ for every $c \in C$: $c(u_0) = 0$ and $d(u_0) = 0$. For every $i, j \in \{1, \ldots, m\}$ and $c \in C$, we set $c(u_i) = 0$ if $c \in B_i$; otherwise, let $c(u_i) = i$. Set $d(u_i) = i$ and $R = C \cup \{d\}$.

From Lemma 1, we can obtain an algorithm with polynomial time complexity for finding a consistent decision table from a given Sperner system K such that $K_d^{-1} = K$. We consider two cases:

If $|A| \leq k$ and $A \not\subset B_i$, for $i = 1, \ldots, m$, then from the definition of anti-keys set, Sperner system, Lemma 1, and the definition of relative reduct, we can see that A must contain at least one reduct of decision table DS. It follows that A is a relative reduct of DS. Otherwise, if A is a relative reduct. From definitions of relative reduct, anti-keys set and Lemma 1, we can claim that A is not a subset of B_i (for every $i = 1, \ldots, m$).

From the above analysis, we deduce that A is not a subset of B_i (for every $i = 1, \ldots, m$) if and only if A is a relative reduct of decision table DS, which proves the theorem. □

By using above results, we have the following corollary for finding the smallest reduct set problem.

Corollary 3. *For a given consistent decision table $DS = \big(U, C \cup \{d\}, V, f\big)$, if NP $\neq$ P then there is no polynomial time algorithm to find a reduct set for a decision table DS with the smallest size.*

It is well-known that if the problem class identified by the deterministic Turing machine is P and the problem class identified by the non-deterministic Turing machine is NP, then the NP $\neq$ P problem is one of prominent outstanding unsolved problems in computer science. However, to the best of our knowledge, up to now, it is widely believed that P and NP are distinct classes.

4. Conclusions. We have investigated the computational complexity for attribute reduction-based problems by making use of rough set theory and relational database theory. We have shown that the problems of checking whether an attribute is reducible or redundant can be solved in polynomial time. We have introduced a definition for a relative reduct set, and then presented two NP-complete problems related to cardinality of a relative reduct set.

REFERENCES

[1] Aho A. V., J. E. Hopcroft, J. D. Ullman. The Design and Analysis of Computer Algorithms. Addison-Wesley, 1974.

[2] Bai J., K. Xia, Y. Lin, P. Wu. Attribute Reduction Based on Consistent Covering Rough Set and Its Application. *Complexity*, Hindawi, **1** (2017), 1–9.

[3] Chebrolu S., S. G. Sanjeevi. Attribute Reduction on Continuous Data in Rough Set Theory Using Ant Colony Optimization Metaheuristic. In: Proceedings of the Third International Symposium on Women in Computing and Informatics, Kochi, 2015, 17–24.

[4] Degang C., W. Changzhong, H. Qinghua. A New Approach to Attribute Reduction of Consistent and Inconsistent Covering Decision Systems with Covering Rough Sets. *Information Sciences*, **177** (2007), No 17, 3500–3518.

[5] Demetrovics J., H. M. Quang, V. D. Thi, N. V. Anh. An Efficient Method to Reduce the Size of Consistent Decision Tables. *Acta Cybernetica*, **23** (2018), No 4, 1039–1054.

[6] Demetrovics J., V. D. Thi. Some Remarks on Generating Armstrong and Inferring Functional Dependencies Relation. *Acta Cybernetica*, **12** (1995), No 2, 167–180.

[7] Demetrovics J., V. D. Thi, N. L. Giang. An Efficient Algorithm for Determining the Set of All Reductive Attributes in Incomplete Decision Tables. *Cybernetics and Information Technologies*, **13** (2013), No 4, 118–126.

[8] Demetrovics J., V. D. Thi, N. L. Giang. On Finding All Reducts of Consistent Decision Tables. *Cybernetics and Information Technologies*, **14** (2014), No 4, 3–10.

[9] Demetrovics J., V. D. Thi, N. L. Giang, D. T. Huy. On the Time Complexity of the Problem Related to Reducts of Consistent Decision Tables. *Serdica Journal of Computing*, **9** (2015), No 2, 167–176.

[10] Ghosh S., R. Biswas, S. S. Alam. Reduction of the Decision Table: A Rough Approach. *International Journal of Intelligent Systems*, **19** (2004), No 12, 1143–1150.

[11] Giang N. L., J. Demetrovics, V. D. Thi, P. D. Khoa. Some Properties Related to Reduct of Consistent Decision Systems. *Cybernetics and Information Technologies*, **21** (2021), No 2, 3–9.

[12] Giang N. L., V. D. Thi. Some Problems Concerning Condition Attributes and Reducts in Decision Tables. In: Proceeding of the Fifth National Symposium "Fundamental and Applied Information Technology Research" (FAIR), Dong Nai, Viet Nam, 2011, 142–152.

[13] Li H., W. Zhang, P. Xu, H. Wang. Rough Set Attribute Reduction in Decision Systems. In: G.-Y. Wang, J. F. Peters, A. Skowron, Y. Yao (eds). Rough Sets and Knowledge Technology. *Lecture Notes in Computer Science*, **4062** (2006), 135–140.

[14] Mi J.-S., W.-Z. Wu, W.-X. Zhang. Approaches to Knowledge Reduction Based on Variable Precision Rough Set Model. *Information Sciences*, **159** (2004), No 3, 255–272.

[15] Pawlak Z. Rough Sets. *International Journal of Computer & Information Sciences*, **11** (1982), No 5, 341–356.

[16] Pawlak Z. Rough Sets: Theoretical Aspects of Reasoning about Data. Springer, Netherlands, 1991.

[17] Pawlak Z., A. Skowron. Rudiments of Rough Sets. *Information Sciences*, **177** (2007), No 1, 3–27.

[18] QIAN Y., J. LIANG, W. PEDRYCZ, C. DANG. Positive Approximation: An Accelerator for Attribute Reduction in Rough Set Theory. *Artificial Intelligence*, **174** (2010), No 9, 597–618.

[19] QIAN Y., J. LIANG, W. PEDRYCZ, C. DANG. Incremental Feature Selection for Dynamic Hybrid Data Using Neighborhood Rough Set. *Knowledge-Based Systems*, **194** (2020), 105516.

[20] THI V. D., N. L. GIANG. A Method for Extracting Knowledge from Decision Tables in Terms of Functional Dependencies. *Cybernetics and Information Technologies*, **13** (2013), No 1, 73–82.

[21] VAN D. V., N. L. GIANG, V. D. THI. Generalized Discernibility Function Based Attribute Reduction in Incomplete Decision Systems. *Serdica Journal of Computing*, **7** (2013), No 7, 375–388.

[22] ZHENG K., J. HU, Z. ZHAN, J. MA, J. QI. An Enhancement for Heuristic Attribute Reduction Algorithm in Rough Set. *Expert Systems with Applications*, **41** (2014), No 15, 6748–6754.

Phan Dang Khoa, Vu Duc Thi
Information Technology Institute
Vietnam National University, Hanoi
144, Xuan Thuy, Cau Giay District
Hanoi, Vietnam
e-mail: `{khoapd, vdthi}@vnu.edu.vn`

János Demetrovics
Institute for Computer Science and Control (SZTAKI)
Hungarian Academy of Sciences
13-17, Kende u.
H-1111 Budapest, Hungary
e-mail: `demetrovics@sztaki.mta.hu`

Pham Viet Anh
Hanoi University of Industry
298, Cau Dien St, Bac Tu Liem District
Hanoi, Vietnam
e-mail: `anhpv@haui.edu.vn`

Received March 22, 2021
Final Accepted September 10, 2021